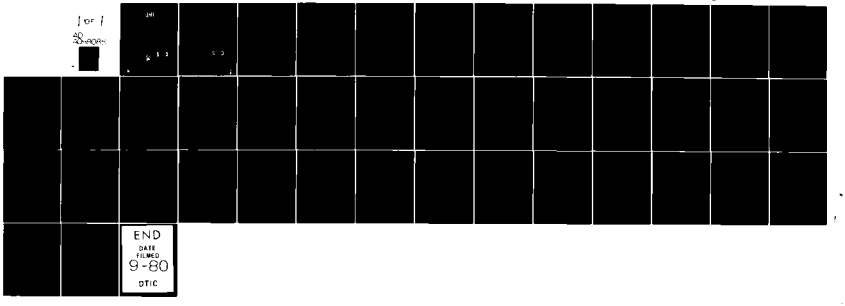
AD-A088 088 NAVAL RESEARCH LAB WASHINGTON DC F/G 9/5
BUILDING BLOCK FOR AN ORTHONORMAL-LATTICE-FILTER ADAPTIVE NETWO--ETC(U)
JUL 80 W F GABRIEL
UNCLASSIFIED NRL-8409 NL

1 of 1
AD
A088088



END
DATE FILMED
9-80
DTIC

NRL Report 8409

Building Block for an Orthonormal-Lattice-Filter Adaptive Network

W. F. GABRIEL

Antenna Systems Staff
Radar Division

July 10, 1980

NAVAL RESEARCH LABORATORY
Washington, D.C.

(11) RR0210 1,
RR0210 5

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: NRL Report 8409	2. GOVT ACCESSION NO.: AD-A088088
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): BUILDING BLOCK FOR AN ORTHONORMAL-LATTICE-FILTER ADAPTIVE NETWORK	5. TYPE OF REPORT & PERIOD COVERED: Interim report on a continuing NRL problem
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): W. F. Gabriel	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, DC 20375	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61150N, RR021 01 01, 0623-0; 61153N, RR021 05 41, 0624-0
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Air Systems Command, Washington, DC 20361 and Office of Naval Research, Arlington, VA 22217	12. REPORT DATE: July 1980
	13. NUMBER OF PAGES: 41
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC ELECTE AUG 19 1980 S D B

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive arrays
Antennas
Adaptive filters
Nonlinear processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The recent algorithm for a multistage multichannel orthonormal lattice filter proposed by M. Aftab Alam is a welcome addition to the library of adaptive-processing algorithms and provides a flexible alternative to the conventional approach of an optimum Weiner filter. His algorithm is based on a Gram-Schmidt orthonormalization procedure which is similar to cascade adaptive processing techniques described in earlier work by B. L. Lewis and F. F. Kretschmer, Jr., and also by L. E. Brennan, J. D. Mallett, and I. S. Reed. One of the most desirable features of this type of processing

(Continues)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

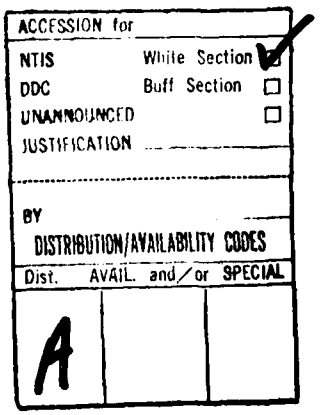
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251950

20. Abstract (Continued)

network is that it can be implemented with simple one-stage orthogonal-filter building blocks which directly filter the input data samples. These building blocks are the major subject of this report, and a particular configuration is developed based on a modified version of the familiar Howells-Applebaum algorithm. It can be implemented in either analog or digital form, data storage is not required, it is unconditionally stable, speed of convergence is no longer a problem, and the design is simple. The performance characteristics of a complete orthogonal-lattice-filter network operating in the spacial domain were simulated for example cases of one, two, and three strong incoherent signal sources spaced within a beamwidth for an eight-element linear-array antenna. The adaptive spacial filter patterns and the transient responses demonstrate that the building block has sufficient transient-response speed and control to permit full use of the processing capabilities inherent in a Gram-Schmidt cascade network.

CONTENTS



BUILDING BLOCK FOR AN ORTHONORMAL-LATTICE-FILTER ADAPTIVE NETWORK

INTRODUCTION

This report presents the modification of an old algorithm and its application to a new algorithm in order to achieve a viable alternative to the conventional optimum-Weiner-filter algorithm. The old algorithm is the conventional Howells-Applebaum adaptive-filter algorithm [1], and it is modified by incorporating a dynamic time constant which is a function of input-channel powel level. This modification, when applied to a one-stage orthogonal filter, forms a building block with the following features: it can be implemented in either analog or digital form, data storage is not required, it is unconditionally stable, convergence speed is no longer a problem, steering-control input is permitted, the design is simple, and it has great application flexibility.

The next two sections are devoted to a comprehensive description and analysis of the one-stage orthogonal-filter building block. The discussion deliberately includes a considerable amount of review material because of preparation for its application to a complicated filter network and also because the digital equivalents of the Howells-Applebaum algorithm have received sparse treatment in the literature.

The new algorithm to which the building block is applied is the algorithm for a multistage multichannel orthonormal lattice filter proposed by Alam [2]. He describes a unified recursive technique to estimate the least-squares coefficients of a broad class of linear filter models. In contrast with conventional algorithms, his algorithm does not require the covariance matrix to compute the filter coefficients but instead directly filters the observed data samples. It incorporates a Gram-Schmidt orthonormalization procedure which is similar to cascade adaptive processing networks described in earlier work by Lewis and Kretschmer [3] and also by Brennan, Mallett, and Reed [4]. The fourth section reviews Alam's algorithm briefly and develops the network transfer matrix, which is necessary to show the equivalence to the optimum Weiner filter.

A Gram-Schmidt orthogonal lattice filter of just one error output is then selected as a demonstration vehicle for the building block, and the fifth section illustrates its spacial-filter performance characteristics for simulated example cases of zero, one, two, and three strong incoherent signal sources spaced within a beamwidth of an eight-element linear-array antenna. For comparison the transient performance of the familiar Howells-Applebaum algorithm on the same data samples is included.

Manuscript submitted March 19, 1980.

ONE-STAGE ORTHONORMAL FILTER

First a one-stage orthonormal filter [2] will be briefly reviewed to establish the concepts of the optimum weight and its relationship to orthogonalization of outputs. This also provides background for the filter networks which will be constructed from the one-stage building block.

Consider the simplified sketch of a one-stage filter shown in Fig. 1. Let the vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$ represent the sequences of discrete-time-sampled voltages from input channels 1 and 2, so that there are two m-dimensional observation vectors:

$$\begin{aligned} \mathbf{V}_1 &= [V_1(1), V_1(2), V_1(3), ..., V_1(i), ..., V_1(m)]^t, \\ \mathbf{V}_2 &= [V_2(1), V_2(2), V_2(3), ..., V_2(i), ..., V_2(m)]^t. \end{aligned} \tag{1}$$

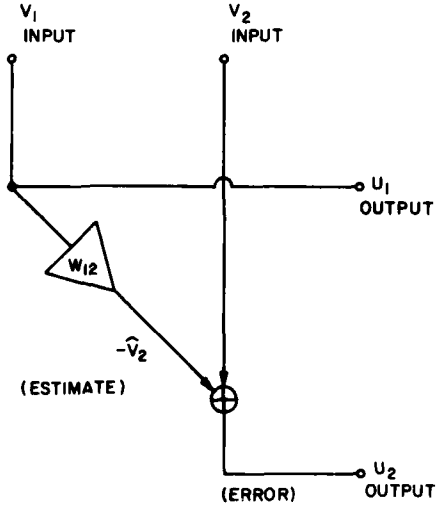


Fig. 1 — One-stage orthonormal filter

One of these vectors, $\mathbf{V}_2$, is chosen to be the *desired vector*, and the remaining vector $\mathbf{V}_1$ is defined as the only vector in the *estimation subspace*. The multiplication of $\mathbf{V}_1$ by a weight coefficient W_{12} then becomes the *estimate vector*, $\hat{\mathbf{V}}_2$:

$$\hat{\mathbf{V}}_2 = - W_{12} \mathbf{V}_1. \tag{2}$$

Also, an *error vector* $\mathbf{U}_2$ is defined as the difference between the desired vector and the estimate vector:

$$\mathbf{U}_2 = \mathbf{V}_2 - \hat{\mathbf{V}}_2. \tag{3}$$

In linear estimation theory a measure of performance often used is a *loss function* consisting of the sum of the squares of the m error terms, that is, the Hermitian vector length of $\mathbf{U}_2$:

$$L(\mathbf{U}_2) = \mathbf{U}_2^{*t} \cdot \mathbf{U}_2 = (\mathbf{V}_2 + W_{12}\mathbf{V}_1)^{*t} \cdot (\mathbf{V}_2 + W_{12}\mathbf{V}_1). \tag{4}$$

The "best performance" or smallest estimation error then consists of minimizing Eq. (4) by taking the derivative of the loss function with respect to the single weight coefficient and setting it equal to zero, whereupon the optimum or least-squares weight value is obtained:

$$W_{12} = -\frac{\mathbf{V}_1^{*t} \cdot \mathbf{V}_2}{\mathbf{V}_1^{*t} \cdot \mathbf{V}_1} = -\frac{\sum_{i=1}^{m} V_1^*(i)\, V_2(i)}{\sum_{i=1}^{m} V_1^*(i)\, V_1(i)}. \tag{5}$$

Equation (5) shows that this is the same weight coefficient as obtained in the familiar Gram-Schmidt orthogonalization procedure [5], wherein one wishes to obtain a vector $\mathbf{U}_2$ which is orthogonal to $\mathbf{V}_1$. Thus the weight defined by Eq. (5) results in a one-stage Gram-Schmidt orthogonal filter which also happens to be an optimal estimation filter. This is in agreement with the orthogonal projection theorem used in optimization theory [6], which states that the desired vector uniquely decomposes into two orthogonal components: the error vector and the estimate vector, where the estimate vector is the orthogonal projection of the desired vector onto the estimation subspace.

The consequent orthogonality of the error $\mathbf{U}_2$ and the estimate $\hat{\mathbf{V}}_2$ is readily shown:

$$\begin{aligned} \hat{\mathbf{V}}_2^{*t} \cdot \mathbf{U}_2 &= \left(\hat{\mathbf{V}}_2^{*t} \cdot \mathbf{V}_2\right) - \left(\hat{\mathbf{V}}_2^{*t} \cdot \hat{\mathbf{V}}_2\right) \\ &= -W_{12}^* \left(\mathbf{V}_1^{*t} \cdot \mathbf{V}_2\right) + W_{12}^* \left(\mathbf{V}_1^{*t} \cdot \mathbf{V}_2\right) \\ &= 0. \end{aligned} \tag{6}$$

In addition, the error vector is orthogonal to the input vector $\mathbf{V}_1$:

$$\begin{aligned} \mathbf{V}_1^{*t} \cdot \mathbf{U}_2 &= \left(\mathbf{V}_1^{*t} \cdot \mathbf{V}_2\right) - \left(\mathbf{V}_1^{*t} \cdot \hat{\mathbf{V}}_2\right) \\ &= -W_{12} \left(\mathbf{V}_1^{*t} \cdot \mathbf{V}_1\right) + W_{12} \left(\mathbf{V}_1^{*t} \cdot \mathbf{V}_1\right) \\ &= 0. \end{aligned} \tag{7}$$

To complete this one-stage filter, an output vector U_1 is defined which is equal to V_1 and therefore orthogonal to the error output U_2,

$$U_1 = V_1, \tag{8}$$

such that now defined are orthogonal vector outputs with the matrix relationship

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ W_{12} & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} . \tag{9}$$

The outputs may be further transformed to have a norm of unity if desired, but this will not be necessary for the networks considered herein.

This filter is a true linear prediction filter [7], because it is attempting to form an estimate of the input signal V_2 in Eq. (2) with its single degree of freedom represented by the weight coefficient W_{12}. The quality of that estimate will depend in part on whether or not the data sequences of m samples represent adequate sampling of the input signal waveforms.

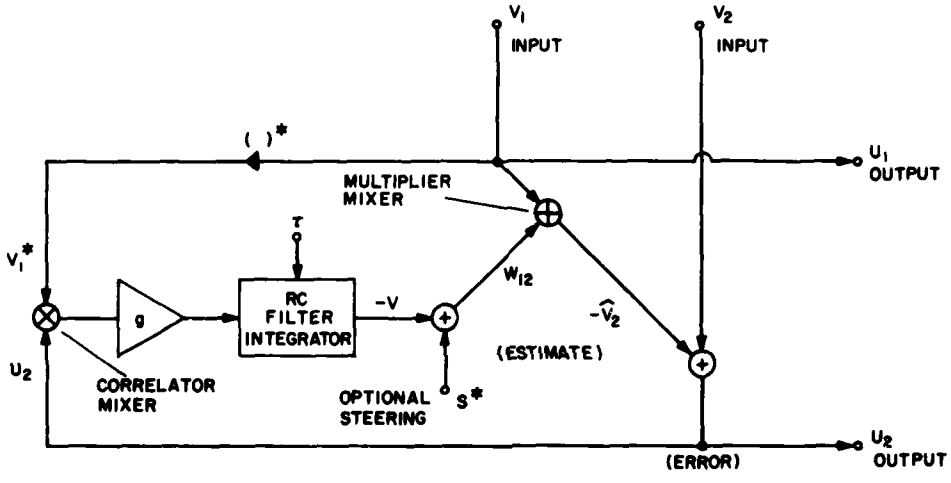
This one-stage filter constitutes a building block for constructing multiple-input optimal filters of greater complexity [2], and the one-stage filter can be implemented in several ways. For example, in a digital system incorporating a computer of sufficient capacity, one would compute the optimum weight directly per Eq. (5) from stored data samples. This is sometimes referred to in the literature as open-loop processing [8] and is generally preferred whenever it can be implemented. However, direct computation is not always cost effective or even possible because of system considerations such as analog component preference, hybrid analog/digital designs, or limited computer capacity. For these cases one may consider implementation of the building block via closed-loop processing, which involves an error-feedback technique. The most practical of these techniques to date is the Howells-Applebaum adaptive algorithm [1], and that will form the basis of the implementation to be described in this report.

ONE-STAGE FILTER USING A FEEDBACK ALGORITHM

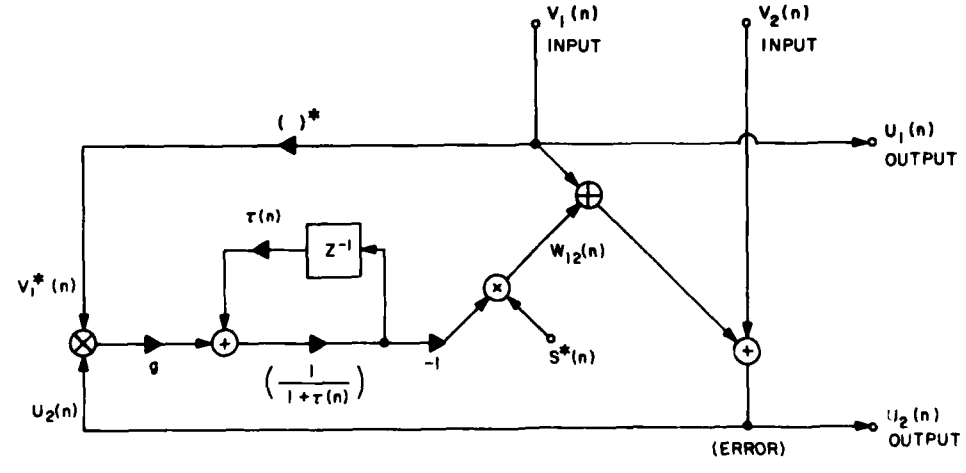
The Howells-Applebaum adaptive algorithm is an ideal candidate for implementing the one-stage orthogonal-filter building block described in the previous section, for the following reasons:

It may be implemented in either analog or digital form;
Data storage is not required;
It is inherently stable and robust;
An option is steering-control input;
It is simple and usually inexpensive;
Performance is usually adequate.

Figure 2 illustrates schematic diagrams for both an analog and a digital recursive version. The convergence speed of this algorithm is sometimes criticized as being too slow for some applications, but a solution for this problem has been developed and will be described as a modification to the basic algorithm.



(a) Analog form

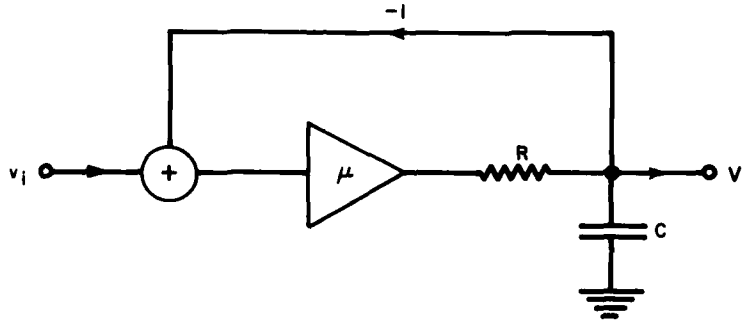


(b) Digital form

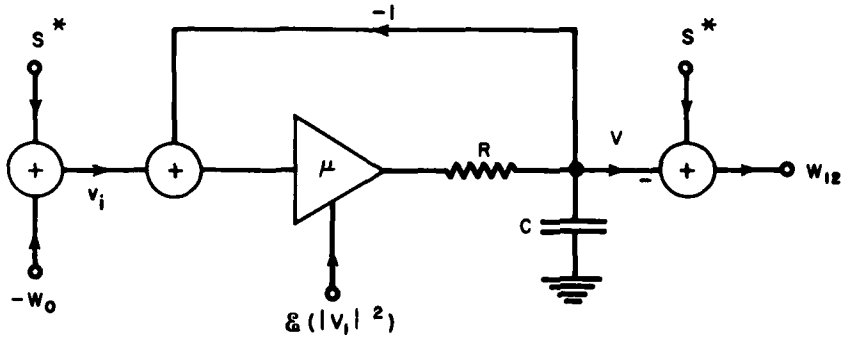
Fig. 2 — One-stage orthogonal filter based on the Howells-Applebaum adaptive feedback algorithm

Analog Implementation

A rather complete discussion of a single Howells-Applebaum analog control loop similar to Fig. 2a is contained in Ref. 9, and it is shown there that this adaptive loop may be represented by the simple first-order follower-servo schematic diagrams illustrated in Fig. 3. That analysis will not be repeated here, but review of a few of the fundamentals is necessary in order to introduce a new dynamic time constant and conversion to a stable digital formulation.



(a) Type-1 follower servo



(b) Equivalent circuit of a one-stage filter

Fig. 3 — Schematic diagrams of feedback servo loops

The differential equation for the RC-filter output voltage V may be written as

$$\tau \frac{dV}{dt} + V = g\left(V_1^* U_2\right), \tag{10}$$

where $\tau = RC$ is the filter time constant, g is a gain constant, and the remaining variables are complex and functions of time. The product within parentheses is the input from the correlator mixer, and it can fluctuate rapidly in accordance with the channel passband, whereas V will fluctuate more slowly in accordance with the filter closed-loop passband. To permit V to remain reasonably independent in a statistical sense from the rapid input signal fluctuations, one generally imposes the restriction that the (two-sided) filter closed-loop passband $2f_3$ must not exceed approximately 10% of the receiver channel passband B_c:

$$\text{restriction } 2\omega_3 \leqslant \frac{2\pi B_c}{10} \quad . \tag{11}$$

The solution of Eq. (10) for the transient behavior of V then requires taking the expected value and assuming a step-function input. The expected value of the correlator input may be written

$$\mathcal{E}\left(V_1^* U_2 \right) = \mathcal{E}\left(V_1^* V_2 \right) + W_{12}\, \mathcal{E}\left(V_1^* V_1 \right) \tag{12}$$

and

$$W_{12} = S^* - V, \tag{13}$$

where S^* is the optional steering weight. Substitution of Eqs. (12) and (13) into Eq. (10) then gives the servo-loop working equation:

$$\tau \frac{dV}{dt} + V = \mu(v_i - V), \tag{14}$$

where

$$v_i = S^* - W_0 \quad \text{(equivalent input)}, \tag{15}$$

$$W_0 = \frac{\mathcal{E}\left(V_1^* V_2\right)}{\mathcal{E}\left(|V_1|^2\right)} \quad \text{(optimum weight)}, \tag{16}$$

$$\mu = g\, \mathcal{E}\left(|V_1|^2 \right) \quad \text{(servo gain factor)}. \tag{17}$$

The expected-value operators in Eqs. (16) and (17) define an average or mean taken over infinite time, such that W_0 is generally referred to as the least-mean-square (LMS) optimum weight. It closely resembles Eq. (5), which represents a summation average taken over a finite number of samples. Under steady-state conditions, under which the first-derivative term in Eq. (14) vanishes, the controlled weight becomes,

$$\text{steady-state } W_{12} = \frac{S^*}{1+\mu} + \frac{\mu}{1+\mu} W_0 . \tag{18}$$

The weight W_{12} approaches W_0 whenever the servo gain factor becomes large ($\mu \gg 1$), thus resulting in the desired orthogonal outputs.

The gain g is adjusted under conditions of quiescent channel receiver noise to result in a minimum value of μ, which is denoted as μ_0 and generally set equal to unity. By introducing a SNR power ratio P_r for channel 1, one can conveniently express μ in terms of μ_0 and P_r:

$$\mu = \mu_0 P_r. \tag{19}$$

Under quiescent noise conditions such that P_r is unity and W_0 is zero (independent channel receiver noise being assumed), Eq. (18) shows that the steady-state value of W_{12} approaches $S^*/(1 + \mu_0)$; that is, it is determined by the steering weight. Thus the controlled weight W_{12} transitions smoothly between the steering value and the optimum value in accordance with the power level of the input signal.

Integration or averaging takes place in accordance with the reciprocal of the closed-loop bandwidth ω_3, which is readily determined from Eq. (14) to be

$$\frac{1}{\omega_3} = \frac{\tau}{1 + \mu} = \frac{1}{\alpha}; \tag{20}$$

that is, the effective averaging decreases as μ increases. This variation with power level creates a problem in selecting a fixed value for τ, because it would have to be chosen with a magnitude large enough to avoid the condition commonly referred to as noisy weights. Substituting restriction (11) into Eq. (20) results in

$$\tau \geqslant \frac{10}{\pi B_c} (1 + \mu), \tag{21}$$

where B_c is the channel bandwidth. Equation (21) shows that τ must be selected for the largest power level to be encountered, and this generally causes it to be much larger than desired at low power levels.

One solution to this problem is to implement a new dynamic time constant τ_d which varies with power level in accordance with the relationship

$$\tau_d = \frac{1}{B_c}\left(\tau_0 + \frac{10}{\pi}\mu\right), \tag{22}$$

where τ_0 is a fixed value selected solely for quiescent noise conditions. The averaging as given by Eq. (20) now becomes

$$\frac{1}{\alpha} = \frac{\tau_0 + \frac{10}{\pi}\mu}{B_c(1 + \mu)}, \tag{23}$$

which never violates the criterion of Eq. (21) and yet becomes no worse than approximately $\tau_0/(1 + \mu_0)$ under quiescent noise conditions. This modification to the basic Howells-Applebaum algorithm could be implemented via monitoring the power (square-law detector) in the V_1 channel and using the resultant current to drive an electronic RC low-pass filter circuit.

Regarding stability, consider Fig. 3a and its differential equation (14). The Laplace transform of an impulse response gives the transfer function for the servo:

$$H(s) = \frac{a}{S + \alpha}, \tag{24}$$

where

$$a = \frac{\alpha\mu}{1 + \mu} \tag{25}$$

and α is defined in Eq. (20). Since α is always a positive real number, the convergence condition for $H(s)$ is always satisfied, and the follower servo is unconditionally stable. The system impulse response will be the inverse Laplace transform of $H(S)$:

$$I(t) = ae^{-\alpha t}, \tag{26}$$

which is the familiar decaying exponential associated with this type of circuit.

Digital Implementation

Consider next the representation of the continuous time signals of the analog circuit with digital equivalents, which are really numerical sequences obtained via periodic sampling as illustrated by the simple sketch in Fig. 4. The nth sample of the sequence of samples may be denoted as $y(n)$:

$$y(n) = y(t)\ \delta(t - nT), \tag{27}$$

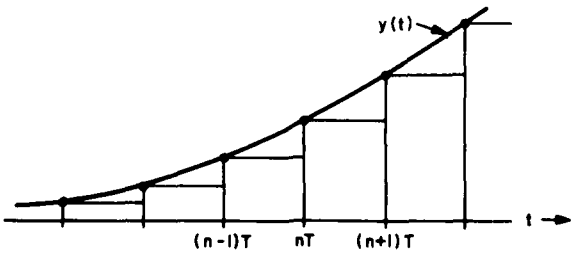


Fig. 4 — Periodic sampling of a continuous function $y(t)$ at instants nT, to obtain an equivalent numerical sequence

where n is an integer, T is the sampling period, and δ is the usual impulse function. The conversion of Eq. (14) to digital form presents an option-choice problem in that there are at least three ways of expressing the first derivative, none of which are precise. With use of Fig. 4, the first derivative may be written

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} \approx \frac{1}{T}\left[y(n+1) - y(n)\right] \qquad \text{(forward difference)},$$

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} \approx \frac{1}{2T}\left[y(n+1) - y(n-1)\right] \quad \text{(central difference)},$$

or

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} \approx \frac{1}{T}\left[y(n) - y(n-1)\right] \qquad \text{(backward difference)}.$$

Obviously these approximations require that $y(t)$ should not have frequency-spectrum components exceeding about 10% of the sampling rate, and preferably much less. The question is whether or not there is a preference in the difference equations, given that the digital processing may involve wide swings in parameter values. Thus the three difference equations were examined, of which the two most important are reported here.

Forward-Difference Digital Diagram

When the first derivative is written as a forward difference, Eq. (14) may be expressed in the form

$$\frac{\tau}{T}V(n+1) = \left(\frac{\tau}{T} - 1\right)V(n) + \mu\left[v_i(n) - V(n)\right], \tag{28}$$

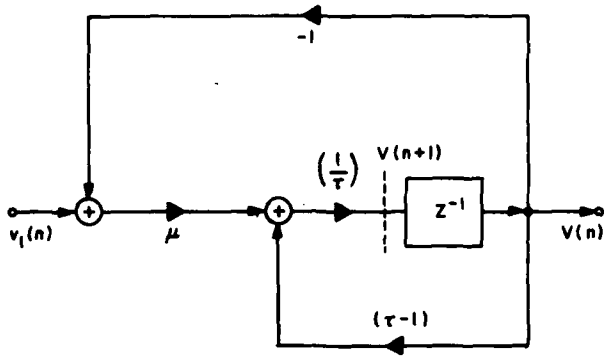
where the next value $V(n+1)$ is calculated on the basis of current values $V(n)$ and $v_i(n)$. This equation leads naturally to the diagram of a forward-difference digital follower servo shown in Fig. 5a. The box labeled Z^{-1} is a standard notation in Z transforms [10] to denote a unit interval delay such that $V(n+1)$ is transformed to $V(n)$ at its output. The recirculation of $V(n)$ through the delay constitutes integration. After Z transforms of Eq. (28) are taken, the servo-system transfer function is found to be

$$H(Z) = \frac{a}{Z - B}, \tag{29}$$

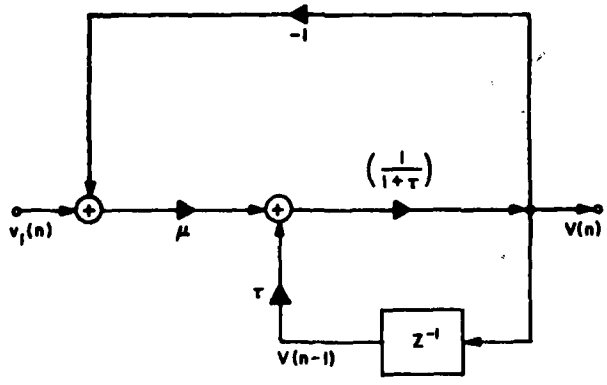
where

$$a = \mu\frac{T}{\tau},$$

$$B = \left[1 - \left(1 + \mu\right)\left(\frac{T}{\tau}\right)\right],$$



(a) Forward-difference derivative



(b) Backward-difference derivative

Fig. 5 — Digital-diagram representation of an analog type-1 follower servo

and

$$Z = \rho e^{j\omega}.$$

When $|\rho| = 1$, Z is constrained to the Z-plane unit circle, where $H(Z)$ becomes $H(\omega)$, and there is a *physical* frequency response or Fourier transform corresponding to the impulse

response of the system. Because of the periodic sampling, ω is a *normalized* frequency which must be expressed in terms of T or the equivalent cutoff frequency f_c:

$$\omega = 2\pi fT = \pi \frac{f}{f_c} . \tag{30}$$

Only under this condition is the analog Fourier transform equal to the discrete-time Fourier transform [10], and it applies only to band-limited functions. The stability and convergence conditions for $H(Z)$ require that

$$|Z| > |B| < 1. \tag{31}$$

This restriction on the value of B is emphasized in the system impulse response derived from the inverse transform of $H(Z)$:

$$h(n) = a\, B^{n-1}\, u(n-1), \tag{32}$$

where $u(n-1)$ is the unit step sequence. Obviously, this exponential sequence will grow unstable if the absolute value of B exceeds unity, so that one must be careful to observe the restriction

$$\left| 1 - (1+\mu) \frac{T}{\tau} \right| < 1. \tag{33}$$

In spite of this danger of instability, the forward-difference digital equivalent of the analog loop has been the digital equivalent most frequently discussed in the literature for both the LMS and the Howells-Applebaum algorithm [11]. The reason for this preference will be discussed in the next main section.

Backward-Difference Digital Diagram

When the first derivative is written as a backward difference, Eq. (14) may be expressed in the form

$$(1 + \tfrac{\tau}{T})\, V(n) = \frac{\tau}{T}\, V(n-1) + \mu\, [v_i(n) - V(n)], \tag{34}$$

where the current value $V(n)$ is calculated on the basis of the previous value $V(n-1)$ and the current input $v_i(n)$. This equation leads naturally to the diagram of a backward-difference digital follower-servo shown in Fig. 5b. The recirculation integrator cannot now become growing and unstable. It can only become decaying for finite values of the time constant. After Z transforms of Eq. (34) are taken, the servo-system transfer function is found to be

$$H(Z) = \frac{aB}{1 - BZ^{-1}} , \tag{35}$$

where

$$B = \frac{1}{1 + a + \frac{T}{\tau}}$$

and a is the same as in Eq. (29). The stability or convergence condition is again given by Eq. (31), but this time B cannot become negative or exceed unity. It cannot even equal unity except for the trivial condition of zero gain. Thus the servo is unconditionally stable for all values of the parameters. This stability is further reflected in the system impulse response derived from the inverse transform of $H(Z)$:

$$h(n) = aB^{n+1}\, u(n), \tag{36}$$

where $u(n)$ is the unit step sequence.

Therefore, the backward-difference digital recursive simulation is the best choice for processing in the digital domain, since it is the only digital simulation option which provides the same unconditional stability as the analog servo circuit. This favorable stability characteristic was recognized by Kretschmer and Lewis [12], who arrived at a backward-difference recursive relationship via an indexing approach. The final digital form represented in Fig. 2b is based on the backward-difference servo diagram of Fig. 5b.

Substituting $v_i = (S^* - W_0)$ into Eq. (34) results in

$$(1 + \mu + \frac{\tau}{T})\, V(n) = \frac{\tau}{T}\, V(n-1) + \mu(S^* - W_0), \tag{37}$$

and a further substitution for V from Eq. (13) results in the backward-difference controlled-weight relationship

$$(1 + \mu + \frac{\tau}{T})\, W_{12}(n) = \frac{\tau}{T}\, W_{12}(n-1) + S^* + \mu W_0. \tag{38}$$

It is of interest to compute $W_{12}(n)$ under quiescent noise conditions wherein W_0 averages to zero and μ becomes μ_0 per Eq. (19):

$$\text{quiescent } W_{12}(n) = \frac{\frac{\tau}{T} W_{12}(n-1)}{1 + \mu_0 + \frac{\tau}{T}} + \frac{S^*}{1 + \mu_0 + \frac{\tau}{T}}, \tag{39}$$

and convergence is reached when $W_{12}(n-1) = S^*/(1+\mu_0)$. This result is identical to that in the analog case. Furthermore, under high-level signal conditions wherein $\mu \gg 1$,

$$\underset{\mu>>1}{W_{12}(n)} = \frac{\frac{\tau}{T}W_{12}(n-1)}{\mu + \frac{\tau}{T}} + \frac{\mu W_0}{\mu + \frac{\tau}{T}}, \tag{40}$$

and convergence is reached when $W_{12}(n-1) = W_0$. Again this result is identical to the analog case.

The closed-loop bandwidth α may be computed from the system transfer function $H(Z)$ in Eq. (35) on the Z-plane unit circle:

$$H(\omega) = \frac{aB}{1 - Be^{-j\omega}}$$

$$= \frac{\mu}{1 + \mu + \frac{\tau}{T}(1 - \cos\omega) + j\frac{\tau}{T}\sin\omega}. \tag{41}$$

If we operate sufficiently below the sampling rate such that $\sin\omega \approx \omega$, then the 3-dB point occurs approximately at

$$\alpha = \omega|_{3\text{ dB}} \approx \frac{1+\mu}{\frac{\tau}{T}}, \tag{42}$$

which is similar to Eq. (20) in the analog case. The noisy-weight criterion then gives an expression similar to Eq. (21) in the analog case,

$$\frac{\tau}{T} > \frac{10}{\pi}(1+\mu), \tag{43}$$

because the IF bandwidth B_c is also the sampling rate for the digital video. This leads to the same requirement for a time constant which varies with power level as in Eq. (22) in the analog case, and the equivalent digital expression becomes

$$\frac{\tau_d}{T} = \tau_0 + \frac{10}{\pi}\mu. \tag{44}$$

The final recursive equation set is obtained simply by substituting for μ and W_0 their current sequential data values. Although this step may seem abrupt, the reader can verify its basis by going back to Eq. (10) and writing the equivalent digital equation for Fig. 2b per the backward difference,

$$\frac{\tau(n)}{T}\left[V(n) - V(n-1)\right] + V(n) = g\left[V_1^*(n)\, U_2(n)\right], \tag{45}$$

which introduces the current data update of the correlator. A few substitutions then result in the basic backward-difference controlled-weight recursive relationship similar to Eq. (38). The complete recursive set of equations is

$$\left[1 + \mu(n) + \frac{\tau(n)}{T}\right] W_{12}(n) = \frac{\tau(n)}{T} W_{12}(n-1) + S^* + \mu(n)\, W_0(n), \tag{46}$$

where

$$W_0(n) = -\frac{V_1^*(n)\, V_2(n)}{V_1^*(n)\, V_1(n)}, \tag{47}$$

$$\mu(n) = g\left[V_1^*(n)\, V_1(n)\right], \tag{48}$$

and

$$\frac{\tau(n)}{T} = \tau_0 + T_f \mu(n), \tag{49}$$

in which $T_f \geq 10/\pi$ (high-power fast time constant). The heretofore fixed factor $10/\pi$ in Eq. (49) has been replaced with the selectable variable T_f, which really represents the high-power fast time constant. This increases the flexibility in $\tau(n)$, and an example will be illustrated later in which it was necessary to increase T_f above the minimum value of $10/\pi$. The reader must appreciate the difference in the variables listed in Eqs. (46) through (48); that is, whereas the previous quantities μ and W_0 were steady-state expected values or means which were used to derive and discuss the transient characteristics of the first-order servo loop, the new quantities $\mu(n)$ and $W_0(n)$ are sequential values which change with each new set of incoming data samples $V_1(n)$ and $V_2(n)$.

A recursive relationship like Eq. (46) does not require storage of any incoming data samples. Instead it uses the data samples to compute a weight update increment consisting of the term

$$\frac{\mu(n)\, W_0(n)}{1 + \mu(n) + \frac{\tau(n)}{T}} \approx \begin{cases} \dfrac{W_0(n)}{1 + T_f}, & \text{for } \mu(n) \gg 1, \\ \dfrac{W_0(n)}{2 + T_f + \tau_0}, & \text{for } \mu(n) \approx 1, \end{cases} \tag{50}$$

where the expression is shown evaluated for the two extremes of $\mu(n)$. The update gives equal weighting to incoming data samples if the magnitudes of the samples $V_1(n)$ remain relatively constant. This update increment is added to the previous value of the weight $W_{12}(n-1)$, which is multiplied by a decay factor:

$$\frac{\frac{\tau(n)}{T}}{1+\mu(n)+\frac{\tau(n)}{T}} W_{12}(n-1) = \begin{cases} \dfrac{T_f}{1+T_f} W_{12}(n-1), & \text{for } \mu(n) \gg 1, \\ \dfrac{T_f+\tau_0}{2+T_f+\tau_0} W_{12}(n-1), & \text{for } \mu(n) \approx 1. \end{cases} \tag{51}$$

Thus, the contributions of all previous data samples are summed (stored) in the weight value $W_{12}(n-1)$, but they decay or fade away at the familiar exponential rate of $1/e$ per the equivalent-time-constant period. From Eqs. (23), (42), and (49) the equivalent time constant is

$$\frac{1}{\alpha(n)} = \frac{\tau_0 + T_f \mu(n)}{1+\mu(n)} = \begin{cases} T_f, & \text{for } \mu(n) \gg 1 \\ \dfrac{\tau_0 + T_f}{2}, & \text{for } \mu(n) \approx 1. \end{cases} \tag{52}$$

The full range of values for $1/\alpha$ versus SNR power ratio P_r (Eq. (19)) is plotted in Fig. 6 for typical values of $T_f = 10/\pi$ and $\tau_0 = 200$. The digital integration loop circuit of Fig. 2b behaves in a manner almost identical to the analog RC filter circuit of Fig. 2a under the conditions assumed herein.

The integration action upon the summed weight is tantamount to increasing its system SNR by an amount approximately equal to the ratio of the channel bandwidth to the (two-sided) RC filter bandwidth. From Eqs. (20) and (23) the ratio of those bandwidths is

$$\frac{B_c}{2f_3} = \frac{\pi\tau_0 + 10\mu}{1+\mu} = \begin{cases} 10, & \text{for } \mu \gg 1, \\ \dfrac{\pi\tau_0 + 10}{2}, & \text{for } \mu \approx 1. \end{cases} \tag{53}$$

Appearing here is the lower-limit factor of 10 originally imposed as a bandwidth restriction (Eq. (11)) to avoid noisy weights, and the system SNR gain then increases to approximately $\pi\tau_0/2$ as input power ratio P_r decreases. The equivalent digital expression simply requires replacement of μ with $\mu(n)$.

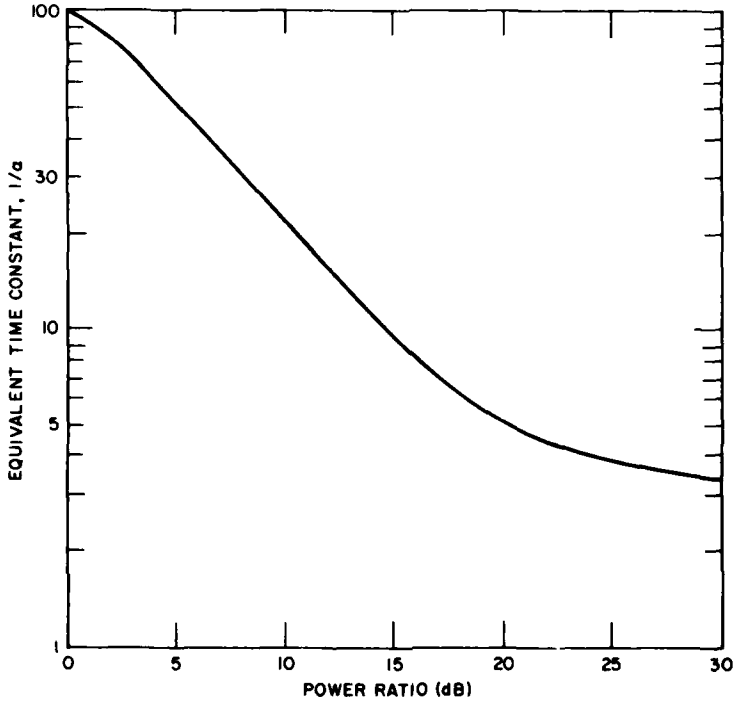


Fig. 6 — Plot of the dynamic equivalent time constant as a function of the power ratio, for typical values $T_f = 3.2$ and $\tau_0 = 200$

MULTICHANNEL FILTERS USING THE BUILDING BLOCK

The one-stage orthogonal-filter building block described in the previous sections may be used in constructing multichannel multistage adaptive filters. Alam [2] stresses the building-block approach, and he proposes a clever vehicle in the form of an orthonormal-lattice-filter algorithm which is a unified recursive technique to estimate the least-squares coefficients of a broad class of linear models. It consists of a sequence of two recursions. First, the Gram-Schmidt orthonormalization procedure estimates a set of inner products from multiple signal vectors, such as multichannel sampled observations. A lattice-type network structure, suitable for adaptive applications, implements the Gram-Schmidt procedure. Second, a recursion is introduced to transform the inner products into model coefficients. In contrast with conventional algorithms, this algorithm does not require the covariance matrix to design the filters but directly filters the observations. He discusses a number of special cases of the multichannel model which appropriately simplify the general structure of the orthonormal-lattice-filter network.

A simplification of Alam's general algorithm lattice network has been chosen as a vehicle for demonstrating the application of our building block. Also, to provide a more familiar multichannel adaptive filter for base line performance comparison, the building

block has been applied to the conventional Howells-Applebaum algorithm, which, like Alam's algorithm, does not require the covariance matrix and directly filters the observations.

Howells-Applebaum Algorithm Filter

The application of our building block to this well-known adaptive algorithm requires that the feedback error signal must now consist of the summation of many adaptively weighted channel signals (the entire array output) instead of just two. The analog form of the algorithm has received such extensive coverage in the literature that it is considered unnecessary to elaborate further herein. For the digital forms, we can derive from Fig. 5 by inspection the two recursive equations for the kth weight of the multichannel filter. The backward-difference form is

$$(1 + \tau)\, W_k(n) = \tau\, W_k(n - 1) + S_k^* - g\, E_k^*(n)\, Y_0(n), \tag{54}$$

and the forward-difference form is

$$\tau\, W_k(n + 1) = (\tau - 1)\, W_k(n) + S_k^* - g\, E_k^*(n)\, Y_0(n), \tag{55}$$

where

$$Y_0(n) = \sum_{k=1}^{K} E_k(n)\, W_k(n) \quad \text{(array output)}. \tag{56}$$

The backward-difference form (Eq. (54)) is unconditionally stable like its analog counterpart, but it shares the same difficulty in solving the simultaneous equations; that is, it requires an orthonormal-mode solution, which becomes a prohibitive computation burden for multiple inputs. The forward-difference form (Eq. (55)), on the other hand, is a set of independent equations which do not require any additional computation burden. Hence the forward-difference form is much preferred for multiple inputs and is the form generally found in the literature [11]. Its instability restrictions are a small price for the convenience of the simpler independent set.

The backward-difference form can also be made independent simply by inserting an additional delay as illustrated in Fig. 7. Equation (54) remains the same for this modification, but the array output changes to

$$Y_0(n) = \sum_{k=1}^{K} E_k(n)\, W_k(n - 1), \tag{57}$$

which causes the feedback error update to become independent of $W_k(n)$. This modification has instability restrictions slightly better than the forward-difference form.

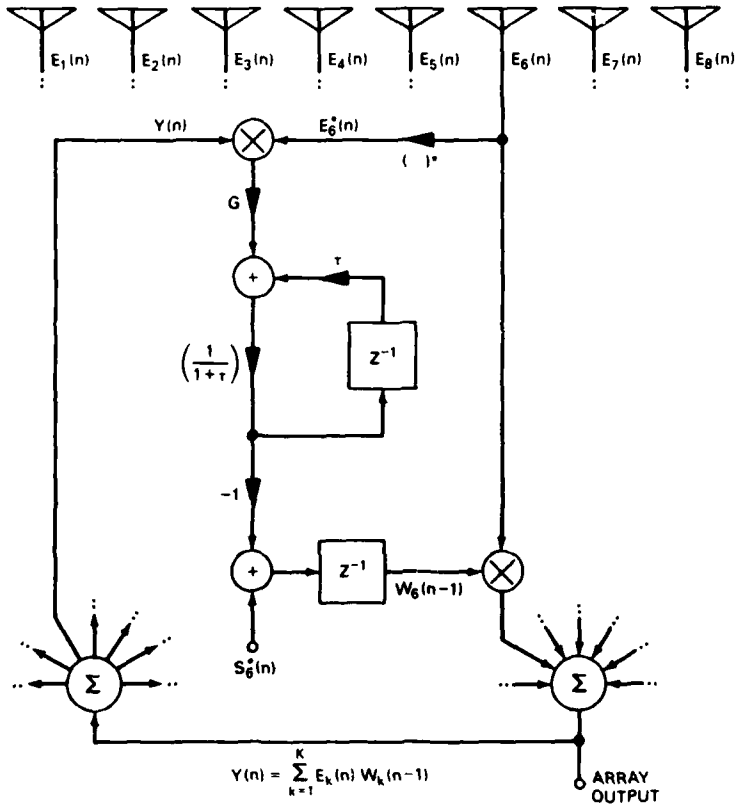


Fig. 7 — Multichannel Howells-Applebaum adaptive algorithm in backward-difference digital recursive form

The new dynamic time constant $\tau(n)/T$ given by Eq. (49) can be used with either form simply by substituting for $\mu(n)$ as follows:

$$\mu(n) = g\left[\mathbf{E}^{*t}(n) \cdot \mathbf{E}(n)\right], \tag{58}$$

where g can be replaced by the quotient of μ_0 divided by the mean channel quiescent noise power as discussed in connection with Eq. (19). The new time constant is valuable for speeding up the convergence rate in comparison to a fixed time constant, but it cannot circumvent the basic eigenvalue transient behavior of a multiple-input Howells-Applebaum-algorithm adaptive filter [9];

The modified backward-difference form is the form used for all reference comparisons of performance in this report and shall be referred to as the baseline Howells-Applebaum algorithm. Its stability restriction is covered by the restriction placed on T_f in Eq. (49): if

the bandwidth criteria for avoiding noisy weights is satisfied, then the system stability criteria is also satisfied.

Orthogonal Lattice Filter of One Error Output

The orthonormal-lattice-filter network of Alam is illustrated in Fig. 8 for the case of two stages or three input channels. It is a complete network in that it transforms K inputs into K orthonormal error outputs where the kth error output is associated with the original kth input. Its transformation action makes it an ideal processing mate for a multiple-beam-antenna spacial-filter system such as a Butler matrix, where one could then obtain an error output for each beam. A desirable feature of Alam's network is that the symmetry causes a large percentage of the coefficients to be identical, thus reducing computation burden. It is instructive to analyze the network associated with just one of these error outputs, because it clearly illustrates implementation of the Gram-Schmidt procedure and formation of the transfer matrix between inputs and outputs. In addition, such a network provides an orthogonal transformation of its own which, though limited and not eigenvector in character, is useful for some adaptive processing situations.

Thus, consider the single-error-output orthogonal-lattice-filter network shown in Fig. 9, which derives directly from Alam's diagram in Fig. 8. Figure 9 also illustrates the earlier Gram-Schmidt cascade processing networks described by Lewis and Kretschmer [3] and Brennan et al. [4]. Although the diagram is drawn to obtain the Kth channel error output, it could just as easily be arranged to obtain the error output from any of the other channels; that is, it is immaterial as to which channel becomes the error output channel. The K channel-input vectors may be represented as a single stacked input vector:

$$\mathbf{V} = \left[\mathbf{V}_1 \; \mathbf{V}_2 \; \mathbf{V}_3 \cdots \mathbf{V}_K \right]^t. \tag{59}$$

Likewise, even though only one unique error output channel has been used, all outputs are represented as the stacked output vector

$$\mathbf{U} = \left[\mathbf{U}_1 \; \mathbf{U}_2 \; \mathbf{U}_3 \cdots \mathbf{U}_K \right]^t, \tag{60}$$

and the transfer matrix $\mathbf{T}$ is sought such that

$$\mathbf{U} = \mathbf{T}\,\mathbf{V}. \tag{61}$$

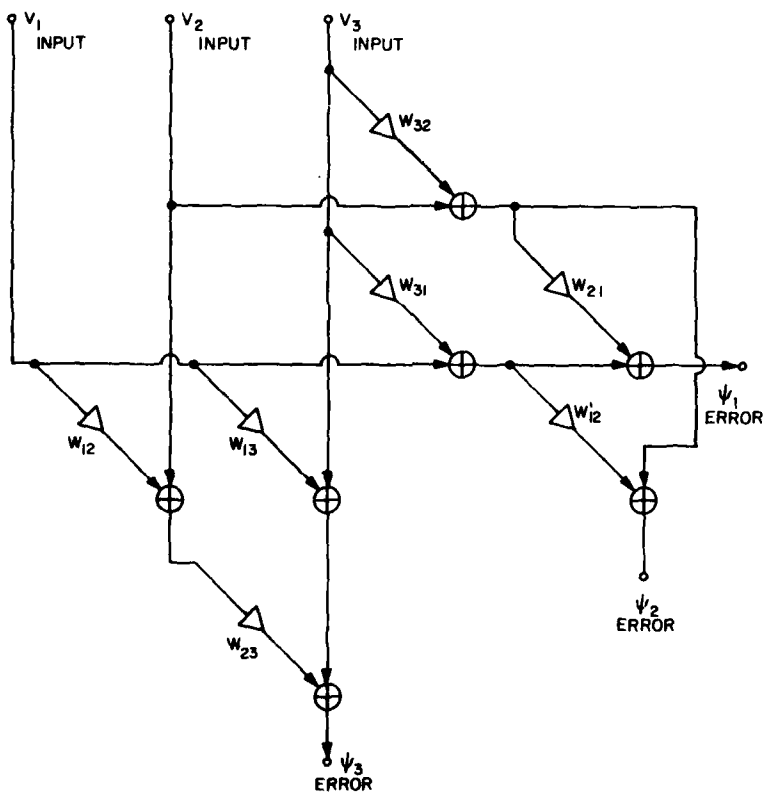


Fig. 8 — Two-stage orthonormal-lattice-filter network of Alam

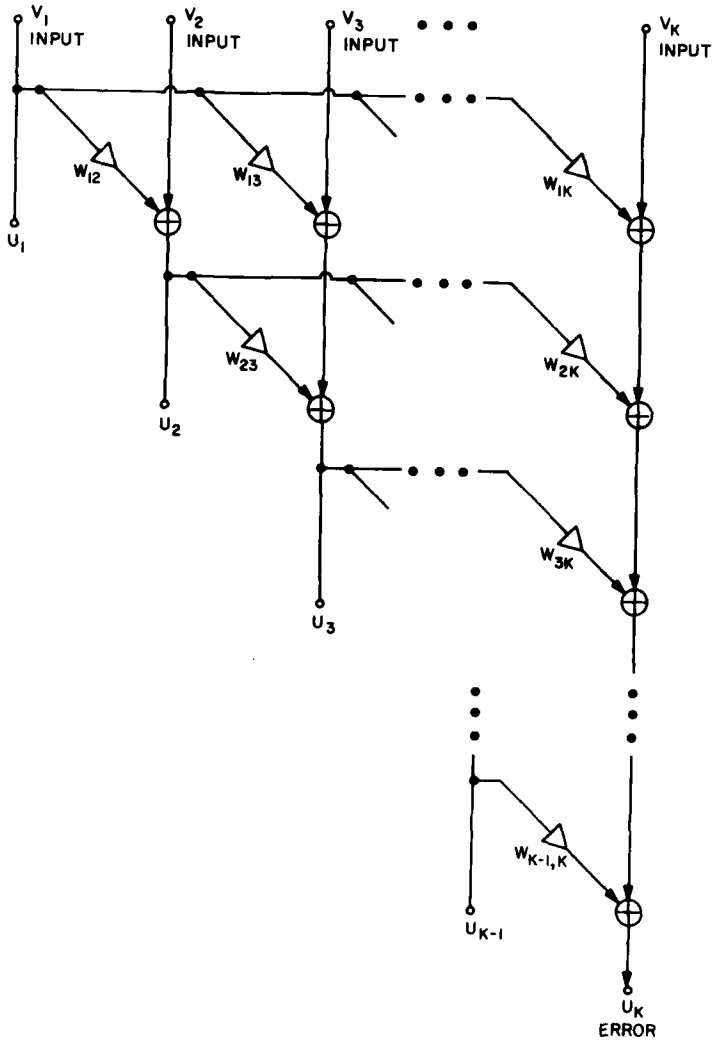


Fig. 9 — *K*-Channel orthogonal-lattice-filter network for single-error output

The transfer matrix will derive from the upper triangular matrix which corresponds to the natural arrangement of weight coefficients in the network This upper triangular matrix can be defined as

$$\begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} & \cdots & W_{1K} \\ 0 & 0 & W_{23} & W_{24} & W_{25} & \cdots & W_{2K} \\ 0 & 0 & 0 & W_{34} & W_{35} & \cdots & W_{3K} \\ 0 & 0 & 0 & 0 & W_{45} & \cdots & W_{4K} \\ 0 & 0 & 0 & 0 & 0 & \cdots & W_{5K} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{62}$$

where each weight value W_{ik} represents the coefficient obtained from the one-stage-filter building block as described in the two preceding main sections. The weight coefficient either may be computed directly in accordance with Eq. (5) for vector inputs of m data samples or may be estimated in real time via the analog/digital network of Fig. 2.

The first two stages in the process are illustrated by the network diagram in Fig. 10, which shows the zero stage and the first stage. The stage will be indicated by an additional subscript on the channel vector such that $\mathbf{V}_{0k}$ represents the zero-stage output vector for the kth channel, $\mathbf{V}_{1k}$ represents the first-stage output vector, $\mathbf{V}_{2k}$ represents the second-stage output vector, etc. Starting with the zero stage, the matrix relationship is

$$\mathbf{V}_0 = \mathbf{A}_0 \, \mathbf{V}, \tag{63}$$

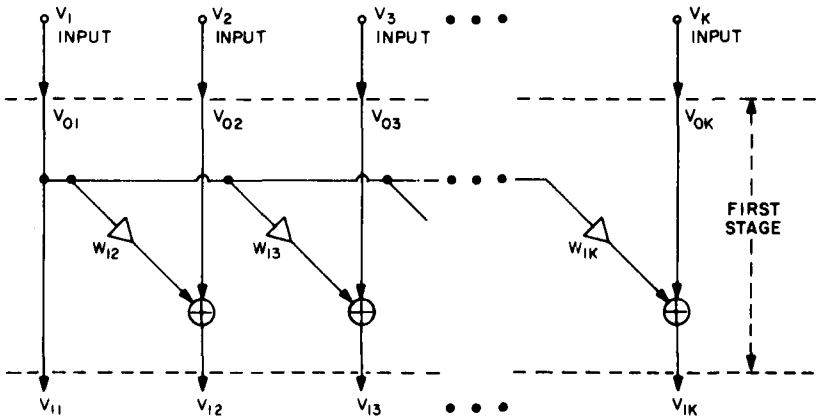


Fig. 10 — First stage of a multistage K-channel orthogonal-lattice-filter network for single-error output

where $A_0 = I$ and I is the identity matrix. Although the transfer matrix A_0 is trivial in this instance, it begins the process. It follows from Fig. 10 and Eq. (9) that the stacked output vector after the first stage may be written

$$V_1 = A_1 V_0, \tag{64}$$

or

$$\begin{bmatrix} V_{11} \\ V_{12} \\ V_{13} \\ \cdots \\ V_{1K} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ W_{12} & 1 & 0 & \cdots & 0 \\ W_{13} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W_{1K} & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} V_{01} \\ V_{02} \\ V_{03} \\ \cdots \\ V_{0K} \end{bmatrix}. \tag{65}$$

If the first-stage transfer matrix of Eq. (65) is defined as A_1, and the total transfer matrix from the original input is defined as T_1, then

$$T_1 = A_1 A_0. \tag{66}$$

When the procedure is followed in this same manner for the second stage, the stacked output vector may be written

$$\begin{bmatrix} V_{21} \\ V_{22} \\ V_{23} \\ \cdots \\ V_{2K} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & W_{23} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & W_{2K} & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} V_{11} \\ V_{12} \\ V_{13} \\ \cdots \\ V_{1K} \end{bmatrix}. \tag{67}$$

If the second-stage transfer matrix of (67) is defined as A_2, and the total transfer matrix from the original input is defined as T_2, then

$$T_2 = A_2 A_1 A_0. \tag{68}$$

This pattern makes it evident that the kth-stage transfer matrix consists of an identity matrix plus the transpose of the kth row of weights taken from the upper triangular matrix of lattice-network weights given in (62). Furthermore the total transfer matrix from the

original input to the output of the kth stage will be the product of all intervening transfer matrices:

$$\mathbf{T}_k = \mathbf{A}_k \, \mathbf{A}_{k-i} \, \mathbf{A}_{k-2} \cdots \mathbf{A}_1 \cdot \mathbf{A}_0. \tag{69}$$

For K input channels, there will be K stages, such that the final transfer matrix from the original input to the final output will consist of a lower triangular matrix equal to

$$\mathbf{T} = \mathbf{A}_{K-1} \, \mathbf{A}_{K-2} \, \mathbf{A}_{K-3} \cdots \mathbf{A}_1 \, \mathbf{A}_0, \tag{70}$$

thereby satisfying the requirement of Eq. (61). Since each stage produces one additional orthogonal output, the final transfer matrix is an orthogonal transformation matrix for the set of input vectors.

The availability of orthogonal outputs from this lattice filter gives the option of easily computed optimum weights, since the sample covariance matrix would be diagonal:

$$\hat{\mathbf{M}}_u = \mathbf{U}^* \, \mathbf{U}^t = \mathbf{T}^* \, \mathbf{V}^* \, \mathbf{V}^t \, \mathbf{T}^t = \mathbf{T}^* \, \mathbf{M} \, \mathbf{T}^t, \tag{71}$$

where M is the sample covariance matrix of the original inputs. Thus

$$\text{optimum weight } \hat{\mathbf{W}}_0 = \hat{\mathbf{M}}_u^{-1} \, \hat{\mathbf{S}}_u^*. \tag{72}$$

To illustrate the equivalence of output and input optimization, one needs only to compute the optimum output, which must be identical for both:

$$\begin{aligned} \text{output} = \mathbf{U}^t \hat{\mathbf{W}}_0 &= \mathbf{V}^t \, \mathbf{T}^t \, \hat{\mathbf{M}}_u^{-1} \, \hat{\mathbf{S}}_u^* \\ &= \mathbf{V}^t \, \mathbf{T}^t \, (\mathbf{T}^* \, \mathbf{M} \mathbf{T}^t)^{-1} \, \mathbf{T}^* \, \mathbf{S}^* \\ &= \mathbf{V}^t \, \mathbf{M}^{-1} \, \mathbf{S}^* \\ &= \mathbf{V}^t \, \mathbf{W}_0, \end{aligned} \tag{73}$$

which assumes that the transfer matrix is nonsingular.

Equation (73) shows that if the optimum output weights $\hat{\mathbf{W}}_0$ are known, then the optimum input weights $\mathbf{W}_0$ may be determined via the relationship

$$\mathbf{W}_0 = \mathbf{T}^t \, \hat{\mathbf{W}}_0. \tag{74}$$

Equation (74) will apply to the transformation of any input/output weight sets, not just the optimum, provided that T is known and nonsingular.

The preceding description illustrates Alam's claim that his algorithm arrives at the optimum filter weights via direct filtering of the observation data and does not require the

covariance matrix or its inverse. The equivalence of his solution to the conventional Weiner filter was shown in Eq. (73). The signal input vector V_K will be estimated from the entire remaining basis set of signal inputs V_1 through V_{K-1} and therefore should be an excellent estimate with little output residual error U_K.

SIMULATION PERFORMANCE CHARACTERISTICS

Some typical performance characteristics for Alam's network of Fig. 9, implemented with the building block of Fig. 2b, have been obtained by simulating its use in a spacial-domain linear prediction filter consisting of a weighted linear array of eight spacial sensor elements, equally spaced. The concept of the filter is illustrated in Fig. 11, where the aim is to estimate the signal arriving at the right-hand unweighted element by subtracting the remaining weighted-element signals and minimizing the output error power. More background on this type of filter is given in Refs. 7 and 13.

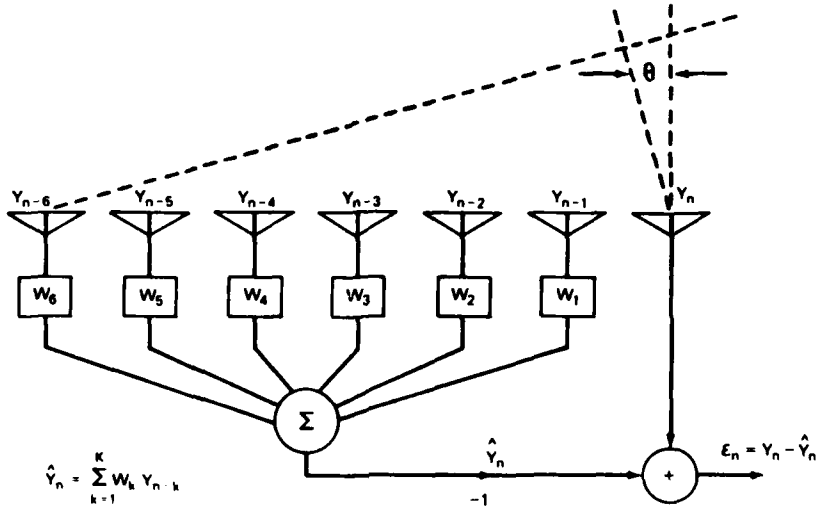


Fig. 11 — Model of the array-aperture linear-prediction spacial filter

The element signal samples are correlated in both space and time, but this is converted to the spacial domain only by the assumption that narrowband filtering precedes the spacial-domain processing.

The nth "snapshot" signal sample at the kth element will consist of independent Gaussian receiver noise η_{kn} plus I incoherent source voltages:

$$E_{kn} = \eta_{kn} + \sum_{i=1}^{I} J_i e^{j(ku_i + \phi_{in})}, \quad 1 \leqslant k \leqslant K, \tag{75}$$

where u_i = $2\pi(d/\lambda) \sin \theta_i$,
d = element spacing, assumed near $\lambda/2$,
λ = wavelength,
θ_i = spacial location angle of the ith source,
J_i = amplitude of the ith source,*
ϕ_{in} = random phase of the nth sample of the ith source,
k = element index,
n = snapshot sample index.

A snapshot is defined as one simultaneous sampling of the aperture signals at all array elements, and it is assumed that N snapshots of data are available.

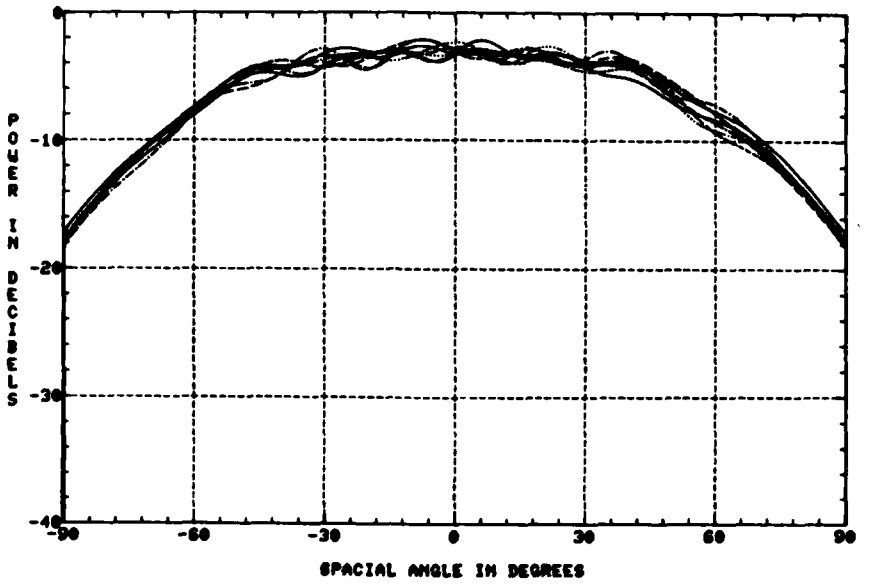
The quiescent spacial pattern for this type of filter is intended to be a wide-beamwidth function, so that the steering vector $\mathbf{S}^*$ is selected to inject a zero weight on every element except the end element:

$$\mathbf{S}^{*t} = [0, 0, 0, 0, 0, 0, 0, 1]. \tag{76}$$

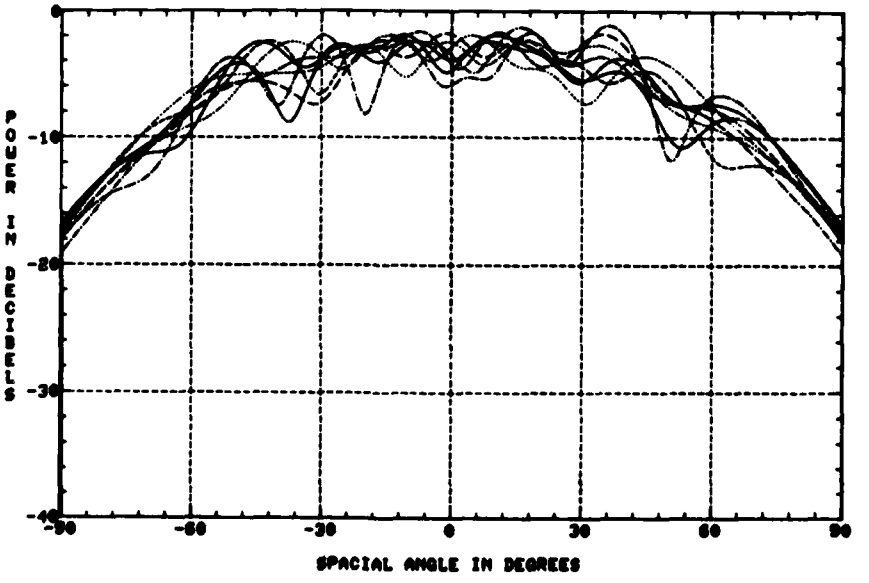
Thus , the quiescent pattern is nominally that of the single end element and will adequately cover the spacial domain between ±90°. In applying this steering vector to Fig. 9, one simply chooses $S^*(n) = 0$ for each building block (Fig. 2b) and notes that the end channel is of unity weight by network construction. Figure 12a illustrates typical quiescent spacial pattern plots for the case in which there are no external sources and the element signals consist only of the independent receiver channel noise. These patterns were computed from weight sets (given by (62)) corresponding to snapshots 200, 400, 500, 600, 700, 800, 900, and 1000. These filter-function patterns always fluctuate with time, and the quiescent fluctuation tolerance is what determines the value of τ_0 in the dynamic time constant of Eq. (49). To directly illustrate the effect of τ_0, Fig. 12b shows the patterns computed from the same snapshot data run as Fig. 12a except that the value of τ_0 was reduced by a factor of 10. The fluctuations of Fig. 12b are considerably greater and would be unacceptable for the present purposes.

Although not shown here, the quiescent pattern fluctuations for the baseline Howells-Applebaum algorithm are similar to Fig. 12a when the same steering vector (Eq. (76)) and same dynamic time constants (Eq. (49)) are used.

*J_i has a constant rather than random amplitude because of a concurrent measurement program involving CW sources sampled at random times.



(a) Quiescent time constant $\tau_0 = 200$



(b) Quiescent time constant $\tau_0 = 20$

Fig. 12 — Typical examples of quiescent patterns for a single-element illumination mainbeam, an eight-element linear array, and no external spacial sources

Single Strong Source

The convergence performance of the orthogonal lattice filter against a single 30-dB source at 21° is shown in Fig. 13a, where the error output power in decibels above receiver noise level is plotted against time measured in snapshot counts. Two symbols are used to plot the error output for each snapshot data sample:

+ = current weights $W_{ik}(n)$ applied to the current input snapshot,
* = previous weights $W_{ik}(n-1)$ applied to the current input snapshot.

This option is available because the building block of Fig. 2b permits application of the current updated weight to the data sample of the current input snapshot. Thus, on the first snapshot the asterisk plots at 30 dB, corresponding to the 30-dB source and quiescent element weights (Eq. (76)), whereas the plus symbol involves the first weight update $W_{ik}(1)$ and is already partially nulling out the source down to 17 dB at the output. Convergence to receiver noise level occurs by the third snapshot, corresponding roughly to the dynamic time constant of Fig. 6 for a 30-dB power ratio. There is not much difference between the two symbol plots after convergence.

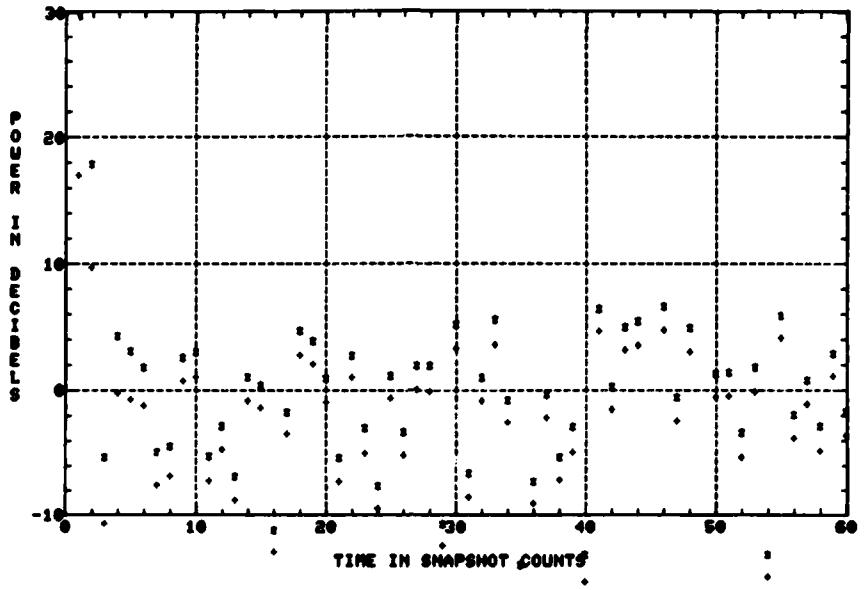
For comparison, the convergence performance of the baseline Howells-Applebaum algorithm against the data samples of the same source is shown in Fig. 13b. The option of the plus-symbol plot is not available with this algorithm because the updated weights cannot be applied to the current snapshot data sample. Instead a continuous-line plot of the rms eigenvalue transient solution has been included based on the averaged sample covariance matrix, with a fixed time constant of 23,600 being used. The asterisks cluster reasonably close to the rms curve for this particular case, as would be expected, and about 20 snapshots are required for this simple algorithm to converge to receiver noise level. After convergence, the asterisks have almost identical plot locations for the two algorithms, indicating that the same optimum-weight solution has been reached.

Figure 14 illustrates a few typical adapted spacial-filter patterns computed from the orthogonal-lattice filter weight sets corresponding to snapshots 3, 10, 30, 90, and 120. The deep null on the single strong source position remains stable, whereas the spacial "passband ripple" will fluctuate like the quiescent patterns of Fig. 12a.

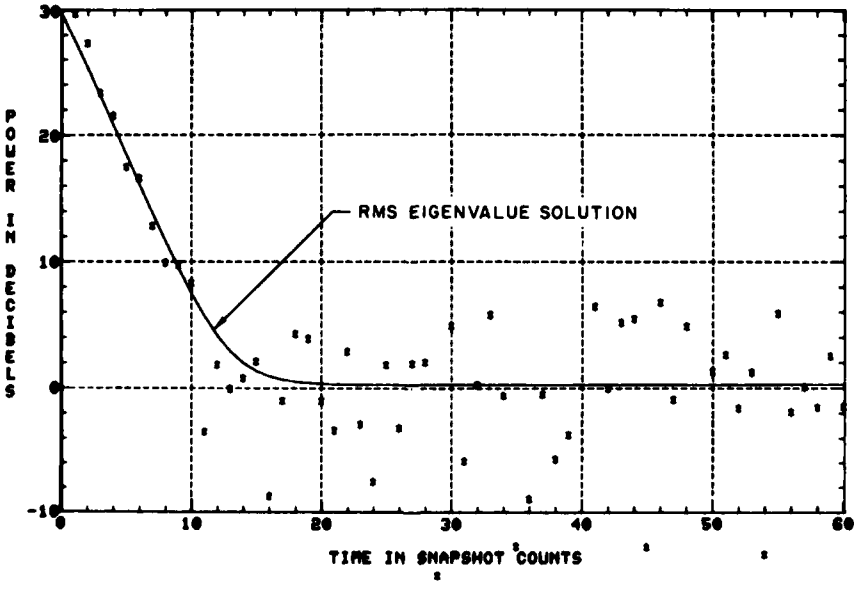
Two Incoherent Strong Sources Closely Spaced

Convergence performance of the orthogonal lattice filter against two incoherent 30-dB sources at 18° and 22° is shown in Fig. 15a. The first snapshot asterisk was at about 35 dB and did not plot here. Convergence to receiver noise level required about seven snapshots for this case.

For comparison, the performance of the baseline Howells-Applebaum is shown in Fig. 15b, which shows that about 60 snapshots are required for convergence. The transient solution for the rms eigenvalue used a fixed time constant of 43,600 and results in a longer convergence time than the new dynamic time constant. The reason for this divergence is that the dynamic time constant takes advantage of instantaneous opposing source signal



(a) Orthogonal-lattice-filter algorithm



(b) Baseline Howells-Applebaum-filter algorithm

Fig. 13 — Time-sequence-snapshot filter error output in decibels above the receiver noise level for a single 30-dB spacial source at 21°

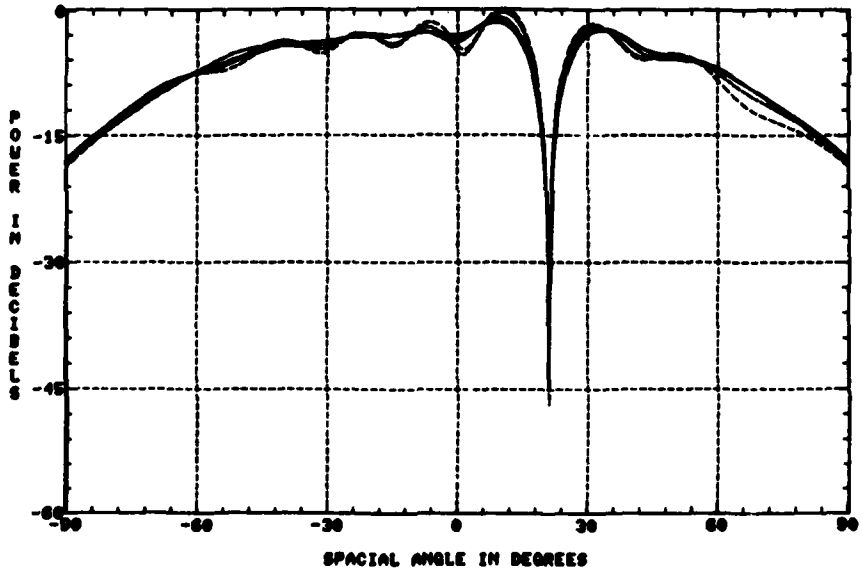
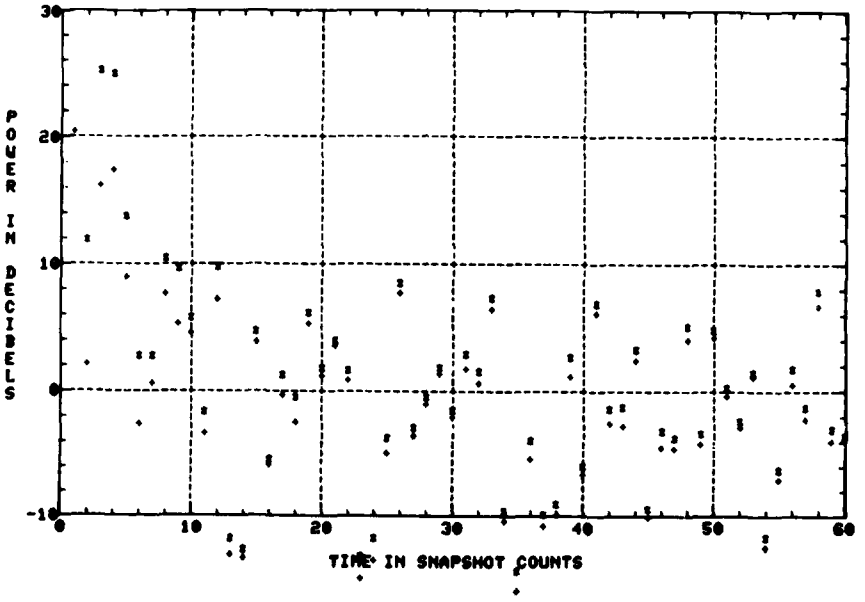
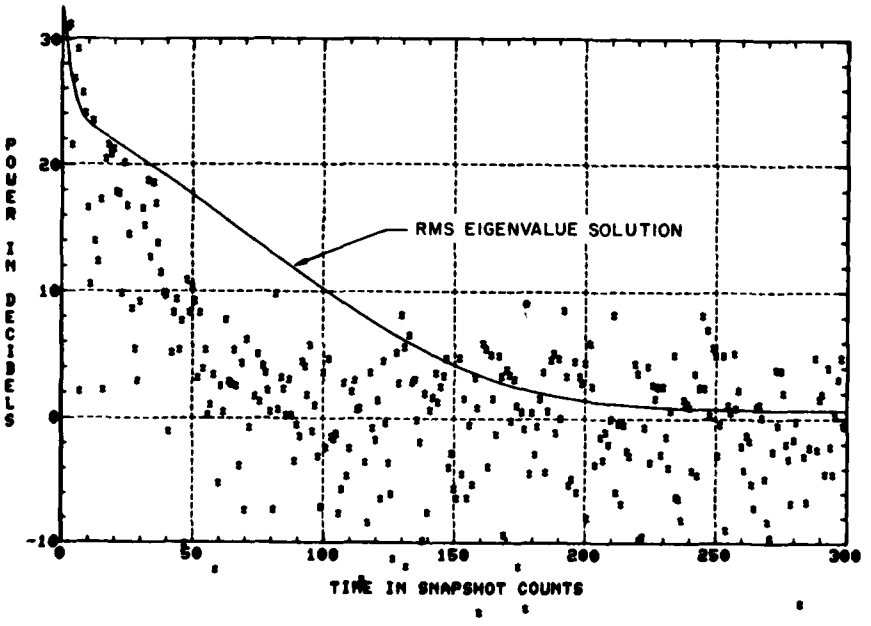


Fig. 14 — Typical examples of spacial-filter patterns computed from orthogonal-lattice-filter weight sets after convergence, for a single 30-dB spacial source at 21°



(a) Orthogonal-lattice-filter algorithm



(b) Baseline Howells-Applebaum-filter algorithm

Fig. 15 — Time-sequence-snapshot filter error output in decibels above the receiver noise level for two 30-dB spacial sources at 18° and 22°

vectors which cause low snapshot power and thus permit a bigger weight update bite. The convergence time has increased so much for this algorithm that it was necessary to increase the time scale.

Figure 16 illustrates a few typical adapted spacial-filter patterns for this case, computed from the orthogonal-lattice-filter weight sets corresponding to snapshots 7, 10, 30, 90, and 121. Two characteristics typical of multiple-source adaptive filtering begin to show here. First, the spacial passband-ripple fluctuations increase in magnitude as the number of degrees of freedom (polynomial zeros) devoted to them decreases. This case has only five out of seven available, whereas the case shown in Fig. 14 has six. Second, the deep nulls on the strong sources require a longer time to become stable, and their fluctuations increase, as the number of captured degrees of freedom increases [9].

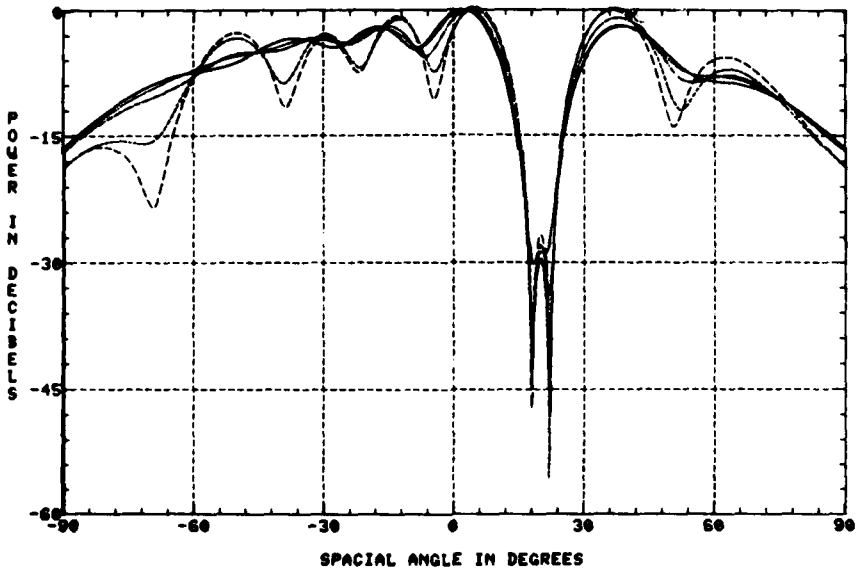
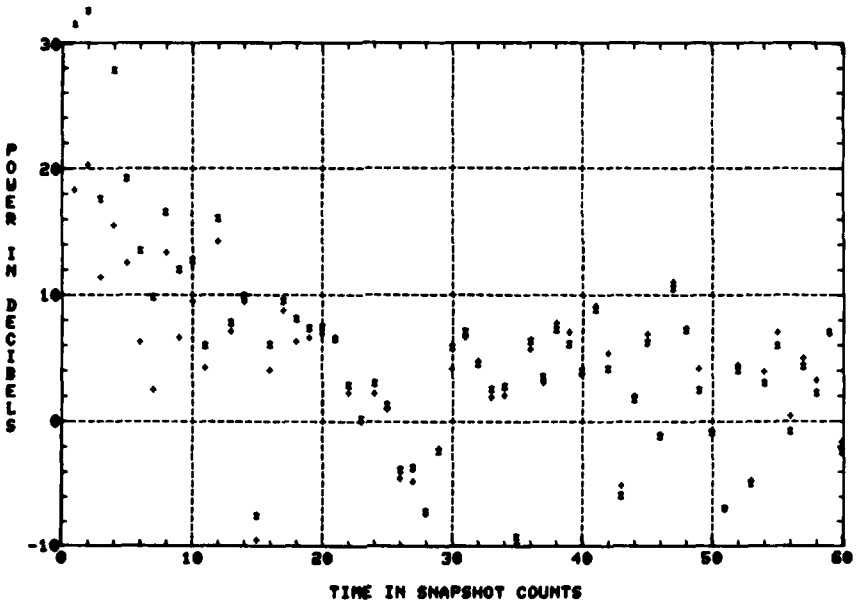


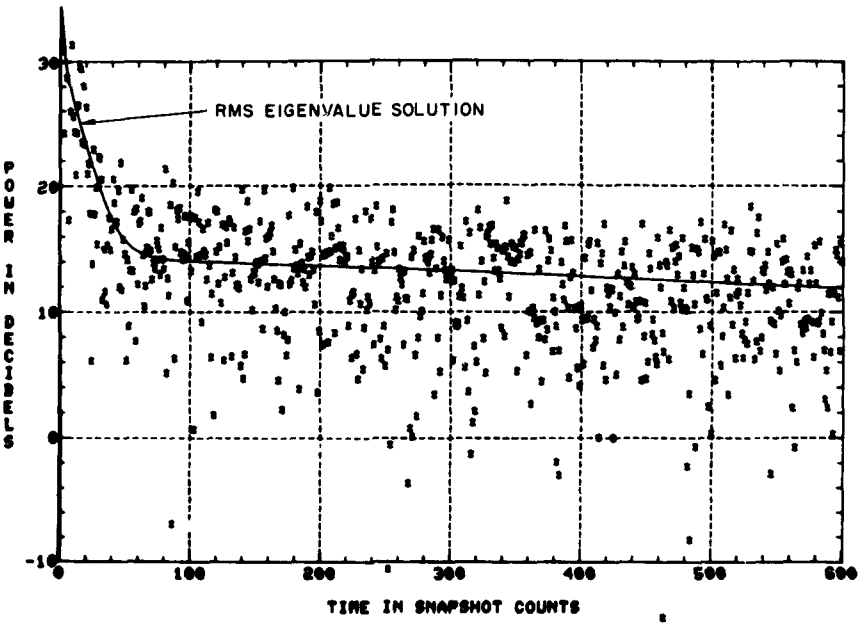
Fig. 16 — Typical examples of spacial-filter patterns computed from orthogonal-lattice-filter weight sets after convergence, for two 30-dB spacial sources at 18° and 22°

Three Incoherent Strong Sources Closely Spaced

Convergence performance of the orthogonal lattice filter against three incoherent 30-dB sources at 14°, 18°, and 22° is shown in Fig. 17a, which shows that about 20 snapshots are required for getting down into the receiver noise level. For comparison, Fig. 17b shows the performance of the baseline Howells-Applebaum algorithm for the same snapshot data samples, and it is obvious that convergence will require thousands of snapshots. One must conclude that the Howells-Applebaum algorithm is unacceptable for cases like this.



(a) Orthogonal-lattice-filter algorithm



(b) Baseline Howells-Applebaum-filter algorithm

Fig. 17 — Time-sequence-snapshot filter error output in decibels above the receiver noise level for three 30-dB spacial sources at 14°, 18°, and 22°

Figure 18 illustrates a few typical adapted spacial filter patterns for this case, computed from the orthogonal-lattice-filter weight sets corresponding to snapshots 10, 30, 60, 90, and 121. The deterioration in passband-ripple fluctuations and the deep-null fluctuations addressed earlier has become unacceptable for locating spacial filter sources even though convergence to the receiver noise level has been satisfactorily achieved.

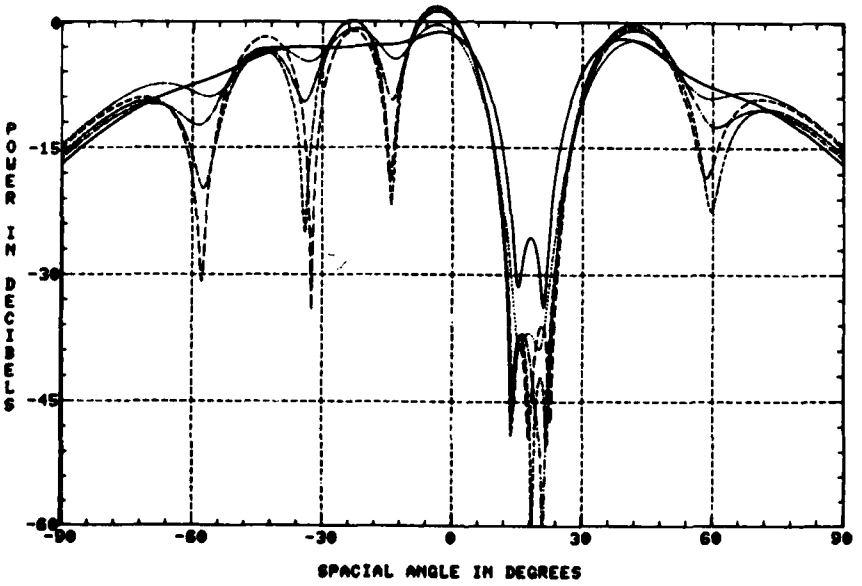


Fig. 18 — Typical examples of spacial-filter patterns computed from orthogonal-lattice-filter weight sets after convergence, for three 30-dB spacial sources at 14°, 18°, and 22°

Fortunately the major problem here is simply one of insufficient integration; that is, the value of $T_f = 3.2$ in the dynamic time constant is too fast and does not provide a sufficient number of data samples for the complicated array-aperture signal waveforms produced by the three simultaneous sources. To directly illustrate this point, Fig. 19 shows a few typical adapted spacial-filter patterns for this same case except that T_f has been increased by a factor of 10 to the value 32. These patterns were computed from the orthogonal-lattice-filter weight sets corresponding to snapshots 200, 300, 400, 500, 700, and 900. Here both the passband-ripple and the deep-null fluctuations have settled down satisfactorily. Of course, the increased value for T_f will result in a slower transient convergence time for the filter amounting to about 70 snapshots for this example. Thus the spacial-filter resolution of multiple sources closely spaced requires a tradeoff between speed of response and pattern stability.

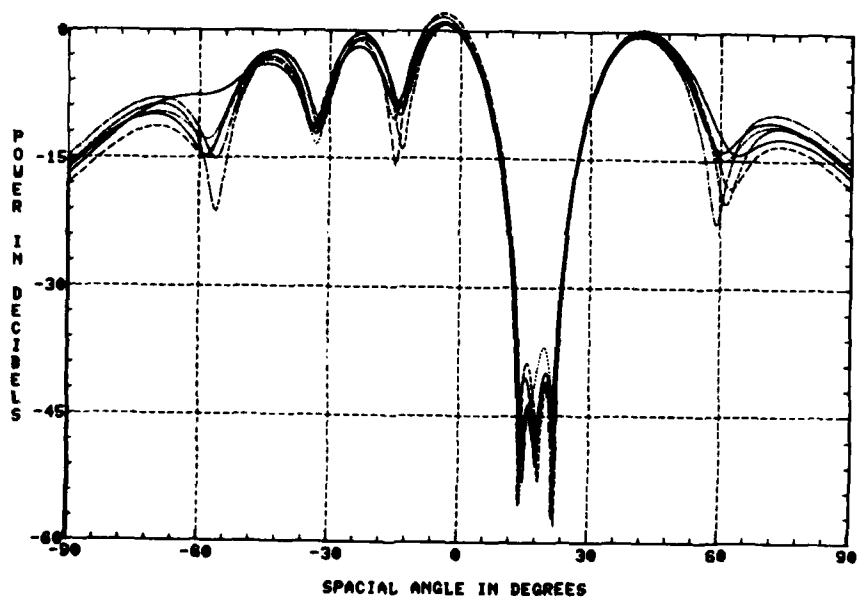


Fig. 19 — Typical examples of spacial-filter patterns computed from orthogonal-lattice-filter weight sets after convergence, with a fast time constant $T_f = 32$, for three 30-dB spacial sources at 14°, 18°, and 22°

CONCLUSIONS

A building block for adaptive filter processors has been described which consists of the Howells-Applebaum algorithm modified to incorporate a dynamic time constant. It can be implemented in either analog or digital form, data storage is not required, it is unconditionally stable, and the dynamic time constant gives it great application flexibility. It offers a viable alternative to direct computation of the optimum weight.

The building block is designed for use in multichannel multistage adaptive filters, and a particular example was chosen from the orthonormal-lattice-filter algorithm proposed by Alam. Performance characteristics for this Gram-Schmidt processor network were simulated in a spacial-domain linear prediction filter using a linear array of eight spacial sensor elements. The transient responses and adaptive spacial filter patterns were computed for one, two, and three strong incoherent sources closely spaced. These simulations demonstrated that the building block has sufficient transient response speed to permit full use of the inherent processing capabilities of such networks.

With this type of network, there is no limitation on convergence speed per se, and convergence time is dominated by other considerations such as the number of degrees of freedom available [11], the number of data samples required by a given signal operating environment, and quiescent system operating requirements. Several examples of these factors were included in the simulations.

REFERENCES

1. S.P. Applebaum, "Adaptive Arrays," IEEE Trans. on Antennas and Propagation 24, 585-598 (Sept. 1976).
2. M.A. Alam, "Orthonormal Lattice Filter—a Multistage, Multichannel Estimation Technique," Geophysics 43, 1368-1383 (Dec. 1978).
3. B.L. Lewis and F.F. Kretschmer, Jr., NRL reports and communications of limited distribution dating back to February 1974.
4. L.E. Brennan, J.D. Mallett, and I.S. Reed, "Convergence Rate in Adaptive Arrays", Technology Service Corp. progress report TSC-PD-A177-2, 15 July 1977, NAVAIR Contract N00019-77-C-0172.
5. F.B. Hildebrand, *Methods of Applied Mathematics*, Prentice-Hall, 1965.
6. D.G. Luenberger, *Optimization by Vector Space Methods*, Wiley, 1969.
7. D.G. Childers, *Modern Spectrum Analysis*, IEEE Press, 1978.
8. F.F. Kretschmer, Jr., and B.L. Lewis, "A Digital Open-Loop Adaptive Processor," IEEE Trans. on Aerospace and Electronic Systems 14, 165-171 (Jan. 1978).
9. W.F. Gabriel, "Adaptive Arrays—An Introduction," Proc. IEEE 64, 239-272 (Feb. 1976).

10. A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, 1975.

11. I.S. Reed, J.O. Mallett, and L.E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," IEEE Trans. on Aerospace and Electronic Systems 10, 853-863 (Nov. 1974).

12. F.F. Kretschmer, Jr. and B.L. Lewis, "An Improved Algorithm for Adaptive Processing," IEEE Trans. on Aerospace and Electronic Systems 14, 172-177 (Jan. 1978).

13. W.F. Gabriel, "Nonlinear Spectral Analysis and Adaptive Array Superresolution Techniques," NRL Report 8345, Feb. 1980.